

LABORATORIO 4

Programas Ada con secuencias y subprogramas

1. Implementa una función llamada **digito_i** que, dado un número de máximo 9 cifras, devuelva la cifra i-ésima empezando a contar por la izquierda. Así Para 846753211 y 1 la función debe devolver 8; y para 846753 y 3 debe devolver 6.
2. Implementa un procedimiento llamado **digito_d** que, dado un número de máximo 9 cifras, devuelva la cifra i-ésima empezando a contar por la derecha. Así Para 846753211 y 1 el procedimiento debe devolver 1; y para 846753 y 3 debe devolver 7.
3. Implementa una función llamada **contar_digitos** que, dado un positivo de máximo 9 cifras, devuelve un Natural indicando cuántas son las cifras que lo constituyen. Así, para un numero como 84567958 debe devolver 8.
4. Implementa el procedimiento **contar_digitos** que, dado un positivo de máximo 9 cifras, devuelve un Natural indicando cuántas son las cifras que lo constituyen. Así, para el número 000009 el resultado debe ser 1.
5. Implementa una función llamada **Iteración_Kaprekar** que dado un número de cuatro cifras (si es menos, se rellena con ceros a la izquierda) realiza las siguientes operaciones: (1) ordena las cifras de menor a mayor formando una cifra; (2) ordena las cifras de mayor a menor formando una segunda cifra; (3) resta el número menor al mayor. Ese es el resultado de la iteración. Por ejemplo, de 4532 se obtiene 3087 (5432-2345); o de 3087 se obtiene 8352 (=8730-0378).
6. Implementa una función llamada **es_capicua** que, dado un número entero positivo, indique si es capicúa. Usa las funciones contar_digitos, y digito_i, y el procedimiento digito_d.
7. Implementa una función en Ada llamada **es_Omirp** que, dado un número entero positivo indique si es Omirp. Un número Omirp es un número primo que, al invertir sus dígitos también es un número primo. Por ejemplo, el 13 y el 1597 son Omirp, ya que son primos y también lo son el 31 y el 7951. Usa la función contar_digitos, y el procedimiento digito_d.
8. Implementa un procedimiento llamado **es_odioso** que indique si un número dado es odioso. Los números odiosos son aquellos que, en su representación binaria, tienen un número impar de unos. Recuerda que para obtener la representación binaria de un número decimal se toman los restos de las sucesivas divisiones entre 2 del número original. El último resto es el bit más significativo (el de la izda) y el primero es el menos significativo (el de la dcha) del número binario.
9. Implementa un programa llamado **lleva_a_kaprekar** que lea números de la entrada y devuelva si, a partir de ese número, se obtiene la constante de Kaprekar (6174) en menos de 100 iteraciones de kaprekar. El resultado del programa debe ser un booleano diciendo si se llega a la constante y en cuantas iteraciones. Utiliza la función Iteración_kaprekar en la resolución.
10. Implementa un subprograma llamado **es_collatz** que compruebe si la secuencia de la entrada se corresponde con la secuencia de Collatz (ver descripción en el enunciado del lab03).
11. Implementa un programa llamado **Pos_Omirp** que, dada una secuencia de números enteros positivos terminada en 0, devuelva el primer número Omirp y su posición en la secuencia (utilizando la función Es_Omirp). Si no hubiese ninguno, los valores que devuelve son dos ceros.
12. Implementa un subprograma llamado **secuencia_odiosos** que, leída una secuencia de números positivos terminada en 0, indique si la suma de los cinco primeros números impares odiosos de la secuencia también es un número odioso. Si la secuencia tiene menos de 5 números impares odiosos, entonces se toman todos los números impares odiosos que tenga la secuencia. Si no hubiera ninguno, la suma es 0. En la implementación se debe utilizar el subprograma es_odioso. Por ejemplo, si la secuencia de entrada es <12, **21**, **25**, **31**, 356, **7**, 2, 5, **13**, 0> (destacados en **negrita** los números impares odiosos) indicará que es TRUE ya que $21+25+31+7+13 = 97$, que tiene un número impar de unos. Si la secuencia de entrada es <13, 7, 0> indicará que es FALSE ya que 20 tiene un número par de unos. Si la secuencia de entrada es <12, 21, 4, 31, 356, 7, 13, 2, 15, 4, 11, 25, 7, 0> indicará que es FALSE ya que $21+31+7+13+11 = 83$, que tiene un número par de unos. Nótese que solo se han recogido los 5 primeros números odiosos ignorando todos los demás.